UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,006	10/22/2003	Tsukasa Takemura	JP920020154US1	5681

25259 7590 07/30/2007

IBM CORPORATION
3039 CORNWALLIS RD.
DEPT. T81 / B503, PO BOX 12195
REASEARCH TRIANGLE PARK, NC 27709

EXAMINER	
HOLLOWAY, DAVID A	
ART UNIT	**PAPER NUMBER**
2109	

NOTIFICATION DATE	DELIVERY MODE
07/30/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

RSWIPLAW@us.ibm.com

Office Action Summary	Application No.	Applicant(s)	
	10/691,006	TAKEMURA, TSUKASA	
	Examiner	Art Unit	
	David A. Holloway	2109	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *22 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *20061109*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-18 are pending in this application.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 3, 5, 15, and 16 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. It is not clear what "the dependency is set in such a way that each task is terminated after awaiting tasks corresponding to all the classes in the interdependence to be ready to terminate" on lines 3-5 of claim 3 means and similarly for claim 15. This phrase is grammatically incorrect and correction is required. The phrase is interpreted as meaning that each task that is waiting for dependent tasks to complete will be executed once all the dependent tasks have been completed. It is not clear what the phrase "past workload achievements in each class are used to estimate the workload of the classes in a multi-regression analysis" on lines 2 and 3 of claim 5 means and similarly for claim 16. The phrase is interpreted to mean that the workload required to implement a class of similar complexity will be used to

estimate the workload required to implement a given class using multi-regression analysis.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-3, and 6-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah (US 6,662,357), hereinafter Bowman in view of Fujikawa et al. (US 6,735,762), hereinafter Fujikawa and further in view of Tim Pyron, "Special Edition Using Microsoft 2000", 2000, QUE, Chapter 1, hereinafter Pyron.

6. As to claim 1, Bowman discloses a scheduling method for scheduling tasks providing a system designed on an object-oriented basis by using a computer (col. 118, lines 13-22, although the system plans a product workload, the scheduling method is applicable for the development of object-oriented software).

7. Bowman discloses setting the dependency between the corresponding tasks on the basis of the dependency (col. 118, lines 61-62, the sequence of job/task is enforced)

8. Bowman does not disclose that the method comprises a first step of extracting classes forming said system from design information thereof, generating tasks corresponding to the extracted classes, and storing information on the generated tasks into a given storage device, a second step of extracting dependency between the classes from the design information of said system, setting the dependency between the corresponding tasks on the basis of the dependency, storing information on the set dependency between the tasks into the given storage device, and a third step of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the information on the dependency between the tasks.

9. However, Fujikawa discloses that the method comprises a first step of extracting classes forming said system from design information (Abstract lines 5-13, the reference examiner examines class definitions and class member definitions), and extracting dependency between the classes from the design information of said system (Abstract, lines 5-7, the reference examiner examines the reference relationships of classes and methods).

10. Bowman and Fujikawa are analogous art because they are both in the same field of endeavor of software development.

11. It would have been obvious to a person having ordinary skill in the art at the time of invention to combine the teachings of Bowman and Fukijawa to extract classes, including dependency information from the design information in order to determine classes that will be influenced by updates to a given class, so that the influenced

classes can be thoroughly tested to ensure that they do not malfunction as a result of the update (Fukijawa, col. 1, lines 14-18, also col. 1, lines 43-50).

12. Neither Bowman nor Fujikawa disclose generating tasks corresponding to the extracted classes. However, in the combined system (a scheduling used for software development), it is inherent that the tasks generated will correspond to the software being developed, i.e., the classes. In the object-oriented system of Fujikawa, the classes are the basic unit of the software under development. It would be obvious to one skilled in the art at the time of invention that in the combined system of Bowman and Fujikawa the tasks generated would correspond to the extracted classes, since the classes need to be implemented in order to develop the software product.

13. Neither Bowman nor Fujikawa disclose storing the generated tasks into a given storage device. However, it would have been obvious to one skilled in the art at the time invention that the generated tasks need to be saved in a storage device in order for them to be useful for project management.

14. Neither Bowman nor Fujikawa disclose storing the information on the set dependency between the tasks into the given storage device. However, for the scheduling information to be useful for a computer-based project management system, the information needs to be stored on a storage device.

15. Neither Bowman nor Fujikawa disclose a third step of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the

information on the dependency between the tasks. However, Pyron discloses that Microsoft project schedules tasks based on information on tasks stored in a storage device, where said information includes dependencies between tasks.

16. Bowman, Fujikawa, and Pyron are analogous art because they are in the same field of endeavor. Bowman and Fujikawa relate to software development and Pyron relates to project management which is pertinent to software development.

17. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman, Fujikawa and Pyron before him or her to schedules tasks based on information on tasks (including dependency information) stored in a storage device, so that the scheduling will be efficient. If a first task, that depends on a second task, needs to be done by a certain time, then the second task must be scheduled in such a way that the first task is not waiting on the dependency. The motivation for storing the information on the tasks in a storage device, would be to allow the information to be easily updated when changes during the project occurs.

18. As to claim 6, Bowman discloses a program product for scheduling tasks providing a system designed on an object-oriented basis by controlling a computer to enable the computer to execute (col. 118, lines 13-22, the scheduling method is applicable for the development of object-oriented software).

19. Bowman discloses setting the dependency between the corresponding tasks on the basis of the dependency (col. 118, lines 61-62, the sequence of job/task is enforced)

20. Bowman does not disclose a first process of extracting classes forming said system from design information thereof, generating tasks corresponding to the extracted classes, and storing information on the generated tasks into a given storage device, a second process of extracting dependency between the classes from the design information of said system, setting the dependency between the corresponding tasks on the basis of the dependency, and storing information on the set dependency between the tasks into the given storage device, and a third process of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the information on the dependency between the tasks.

21. However, Fujikawa discloses a first process of extracting classes forming said system from design information thereof (Abstract, lines 5-13, the reference examiner examines class definitions and class member definitions), and a second process of extracting dependency between the classes from the design information of said system (Abstract, lines 5-7, the reference examiner examines the reference relationships of classes and methods).

22. It would have been obvious to a person having ordinary skill in the art at the time of invention to combine the teachings of Bowman and Fukijawa to extract classes, including dependency information from the design information in order to determine classes that will be influenced by updates to a given class, so that the influenced classes can be thoroughly tested to ensure that they do not malfunction as a result of the update (Fukijawa, col. 1, lines 14-18, also col. 1, lines 43-50).

23. Neither Bowman nor Fujikawa disclose generating tasks corresponding to the

extracted classes. However, in the combined system (a scheduling used for software development), it is inherent that the tasks generated will correspond to the software being developed, i.e., the classes.

24. Neither Bowman nor Fujikawa disclose storing the generated tasks into a given storage device. However, it would be obvious to one skilled in the art that the generated tasks need to be saved in a storage device in order for them to be useful for project management.

25. Neither Bowman nor Fujikawa disclose storing the information on the set dependency between the tasks into the given storage device. However, for the scheduling information to be useful for a computer-based project management system, the information needs to be stored on a storage device.

26. Neither Bowman nor Fujikawa disclose a third process of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the information on the dependency between the tasks. However, Pyron discloses that Microsoft project schedules tasks based on information on tasks stored in a storage device, where said information includes dependencies between tasks.

27. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman and Fujikawa, and the teachings of Pyron before him to schedule the tasks on the basis of the information on the tasks stored in said given storage device and the information on the dependency between the tasks.

The motivation for doing so would have been the same as given above for claim 1.

28. As to claims 2 and 7, neither Bowman nor Fujikawa disclose that if given classes have a relation in which a development of one class requires an earlier development of the other class in said second step, the dependency is set in such a way that a task corresponding to one class concerned is started after termination of a task corresponding to the other class concerned.

29. However, Bowman discloses that tasks that are dependent on other tasks are started after termination of the tasks upon which they are dependent (col. 118, lines 61-62, the tool provides control dependencies, task sequence requirements are enforced).

30. It would have been obvious to one skilled in the art at the time of invention that if the tasks are the development of classes and one class is dependent on the earlier development of another class that the dependent class needs to be developed after the class on which it depends has been developed.

31. As to claims 3 and 8, Bowman discloses that if given classes have an interdependence relation in said second step,the dependency is set in such a way that each task is terminated after awaiting tasks corresponding to all the classes in the interdependence to be ready to terminate (col. 118, lines 61-62, this is the same as claim 2, except there are multiple dependencies).

32. Claims 4 and 5 are rejected under U.S.C. 103(a) as being unpatentable over Bowman in view of Fujikawa, and further in view of Leonard (US 5,729,746).

33. As to claim 4, Bowman discloses a scheduling method for scheduling tasks providing a system designed on an object-oriented basis by using a computer (col. 118, lines 13-22, the scheduling method is applicable for the development of object-oriented software).

34. Bowman does not disclose the scheduling method comprises a first step of extracting classes forming said system from design information thereof, generating tasks corresponding to the extracted classes, and storing information on the generated tasks into a given storage device, a second step of extracting complexity of the classes from the design information of said system, estimating workload required for implementation of the classes on the basis of the complexity, and storing a result of the workload estimation into the given storage device, and a third step of scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the result of the workload estimation.

35. However, Fujikawa discloses that the scheduling method comprises a first step of extracting classes forming said system from design information thereof (Abstract, lines 5-13).

36. Fujikawa discloses a second step of extracting complexity of the classes from the design information of said system (Abstract, lines 5-13, the class definitions and

definitions of the members as well as the class dependencies are extracted and the complexity is dependent on these factors).

37. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman and Fujikawa before him or her to incorporate the class extraction facility of Fujikawa in the system of Bowman in order to be able to better plan for the testing of classes during the development cycle as taught by Fujikawa (col. 1, lines 14-18, col. 1, lines 43-50).

38. Neither Bowman nor Fujikawa disclose that the workload required for implementation of the classes is based on the complexity of the classes. However, Leonard discloses that the complexity of the program logic and the complexity of the data structures are examples of metrics of a software system (col. 3, lines 61-64, it is clear that the metrics listed are directly related to the workload of developing the system. Here the complexity of program logic corresponds to the complexity of the classes).

39. Bowman, Fujikawa, and Leonard are analogous art because they are all in the same field of endeavor of software engineering.

40. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman and Fujikawa, and the teachings of Leonard before him or her to incorporate the metrics of Leonard in estimating the workload required to implement a given task. These metrics are well-known in the art.

The motivation for using the metric of program complexity would have been to allow sufficient time for more complex classes to be implemented. It will obviously take more time to implement a more complex class than a less complex class.

41. Neither Bowman, Fujikawa, nor Leonard disclose storing a result of the workload estimation into the given storage device and scheduling the tasks on the basis of the information on the tasks stored in said given storage device and the result of the workload estimation. However, it would have been obvious to one skilled in the art at the time of invention to store a result of the workload estimation into the given storage device and schedule the tasks on the basis of the information on the tasks stored in said given storage device. The motivation for doing so would have been to allow the scheduling to use information that has already been stored in combination with the workload estimation result to efficiently schedule tasks.

42. As to claim 5, the claim is rejected for the same reasons as claim 4, in addition, Leonard discloses the method according to claim 4, wherein, in said second step, past workload achievements in each class are used to estimate the workload of the classes in a multi-regression analysis (col. 14, lines 48-52, multiple regression analysis of past metrics are used for estimation, here the metric is workload for a class).

43. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman, Fujikawa in view of Pyron, as applied to claim 6 above, and further in view of Leonard.

44. As to claim 9, Fujikawa discloses the program product according to claim 6, wherein said computer is further caused to execute a fourth process of extracting complexity of said classes from said design information of the system (Abstract, lines 5-13, the class definitions and definition of the members as well as the class dependencies are extracted and the complexity is dependent on these factors)

45. Neither Bowman, Fujikawa, nor Pyron discloses estimating workload required for implementation of the classes on the basis of the complexity. However, Leonard discloses estimating workload required for implementation of the classes on the basis of the complexity (col. 3, lines 61-64, the complexity of the program logic and the complexity of the data structures are examples of metrics of a software system, it is clear that the metrics listed are directly related to the workload of developing the system. Here the complexity of program logic corresponds to the complexity of the classes).

46. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman, Fujikawa, Pyron, and Leonard before him to estimate workload required for implementation of the classes on the basis of the complexity and to incorporate the metrics of Leonard in estimating the workload required to implement a given task. These metrics are well-known in the art. The motivation for using the metric of program complexity would have been to allow sufficient time for more complex classes to be implemented. It will obviously take more time to implement a more complex class than a less complex class.

47. Neither Bowman, Fujikawa, Pyron, nor Leonard disclose storing a result of the workload estimation into the given storage device and considering the result of the workload estimation when scheduling the tasks.

48. However, it would have been obvious to one skilled in the art at the time of invention to store a result of the workload estimation into the given storage device and consider the result of the workload estimation when scheduling the tasks. The motivation for doing so would have been to have sufficient knowledge of the tasks available, so that the necessary resources for completion of the task could be allocated. Clearly, tasks with higher workloads will require more resources to be allocated to them.

49. As to claim 10, the claim is rejected for the same reasons as claim 9 above. In addition, Leonard discloses the program product according to claim 9, wherein, in said fourth process of said program, past workload achievements in each class are used to estimate the workload of the classes in a multi-regression analysis (col. 14, lines 48-52, multiple regression analysis of past metrics are used for estimation, here the metric is workload for a class).

50. Claims 13-16 are rejected under 35 USC 103(a) as being unpatentable over Bowman in view of Fujikawa, and Leonard and further in view of Pyron.

51. As to claim 13, Bowman discloses a task scheduling apparatus (col. 118, lines 13-22).

52. Bowman also discloses setting the dependency between the corresponding task on the basis of the dependency (this is part of the dependency setting unit, see col. 118, lines 61-62).

53. Bowman does not disclose that the task scheduling apparatus comprises:
a task generation unit for extracting classes forming a system from design information of the system designed on an object-oriented basis, generating tasks corresponding to the extracted classes,

a dependency setting unit for extracting the dependency of the classes from the design information of said system and setting dependency between the corresponding tasks on the basis of the dependency,

a workload estimation unit for extracting complexity of the classes from the design information of said system and estimating workload required for implementation of the classes on the basis of the complexity, and

a scheduling unit for scheduling the tasks on the basis of the information on the tasks generated by said task generation unit, information on the dependency between the tasks set by said dependency setting unit, and the estimation result made by said workload estimation unit.

54. However, Fujikawa discloses a task generation unit for extracting classes forming a system from design information of the system designed on an object-oriented basis, generating tasks corresponding to the extracted classes (Abstract, lines 5-13, class

information that comprises information on the tasks is extracted from design information).

55. Fujikawa also discloses a dependency setting unit for extracting the dependency of the classes from the design information of said system (Abstract, lines 5-13, the reference examiner is part of the dependency setting unit, the other part is handled by Bowman, col. 118, lines 61-62).

56. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman and Fujikawa before him or her to incorporate the task generation of Fujikawa in the task scheduling apparatus of Bowman. The motivation for doing so would have been to generate tasks related to classes that are updated (Fujikawa, col. 1, lines 14-8, lines 43-50).

57. Neither Bowman nor Fujikawa discloses a workload estimation unit for extracting complexity of the classes from the design information of said system and estimating workload required for implementation of the classes on the basis of the complexity However, Leonard discloses a workload estimation unit for extracting complexity of the classes from the design information of said system and estimating workload required for implementation of the classes on the basis of the complexity (col. 3, lines 61-64).

58. It would have been obvious to a person having ordinary skill in the art at the time of invention to combine the teachings of Bowman and Fujikawa, and the teachings of Leonard by incorporating the workload estimation unit in the combined system of

Bowman and Fujikawa in order for the scheduling unit to have an important metric of the software being developed. This metric is commonly used in scheduling of software development projects.

59. Neither Bowman, Fujikawa, nor Leonard discloses a scheduling unit for scheduling the tasks on the basis of the information on the tasks generated by said task generation unit, information on the dependency between the tasks set by said dependency setting unit, and the estimation result made by said workload estimation unit. However, Pyron discloses a scheduling unit for scheduling the tasks on the basis of the information on the tasks generated by said task generation unit, information on the dependency between the tasks set by said dependency setting unit, and the estimation result made by said workload estimation unit (Microsoft Project generated schedules based on information stored on the tasks. It is clear that dependency information and estimation of workload will be factors used in generating the schedule).

60. Bowman, Fujikawa, Leonard, and Pyron are analogous art because they are in the same field of endeavor. Fujikawa, Leonard, and Pyron relate to software development, and Bowman relates to scheduling and project management which is pertinent to software development.

61. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman, Fujikawa, Leonard, and Pyron before him or her to incorporate the scheduling unit disclosed by Pyron in the combined system of Bowman, Fujikawa, and Leonard. The motivation for doing so would be to automate the scheduling of tasks and to allow a schedule to be easily updated when task information

is updated.

62. As to claim 14, Bowman discloses that the apparatus according to claim 13, wherein, if given classes have a relation in which a development of one class requires an earlier development of the other class, said dependency setting unit sets the dependency in such a way that a task corresponding to one class concerned is started after termination of a task corresponding to the other class concerned (col. 118, lines 61-62).

63. As to claim 15, Bowman discloses that the apparatus according to claim 13, wherein, if given classes have an interdependence relation, said dependency setting unit sets the dependency in such a way that each task is terminated after awaiting tasks corresponding to all the classes in the interdependence to be ready to terminate (col. 118, lines 61-62, this is the same as claim 14 except that there are multiple dependencies).

64. As to claim 16, the claim is rejected for the same reasons as claim 13. In addition, Leonard discloses that the apparatus according to claim 13, wherein said workload estimation unit uses past workload achievements in each class to estimate the workload of the classes in a multi-regression analysis (col. 14, lines 48-52).

65. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman in view of Fujikawa, and Pyron as applied to claim 6 above, and further in view of Narimatus et al. (US 5,826,236), hereinafter Narimatus.

66. As to claim 11, neither Bowman, Fujikawa, nor Pyron discloses that the program product according to claim 6, wherein said computer is further caused to execute a fifth process of generating and outputting a Gantt chart having a description of said task schedule generated in said third process. However, Narimatsu discloses generating and outputting a Gantt chart having a description of said task schedule (Abstract, last sentence, a scheduling chart is generated, Fig. 5-13 display Gantt charts).

67. Bowman, Fujikawa, Pyron, and Narimatsu are analogous art because they are all in the same field of endeavor. Bowman, Fujikawa, and Pyron relate to software development, and Narimatsu relates to scheduling and project management. Scheduling and project management is pertinent to software development.

68. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman, Fujikawa, and Pyron, and the teachings of Narimatsu before him or her to incorporate the Gantt chart producing invention of Narimatsu in the combined system of Bowman, Fujikawa, and Pyron. The motivation for doing so would have been to allow the scheduling unit that has information on the tasks stored in a storage device to generate a scheduling chart that shows dependencies and duration of tasks. Gantt charts are commonly used to give presentations during project meetings.

69. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman in view of Fujikawa, Pyron, as applied to claim 6 above, and further in view of Craig

Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process", Second Edition, July 13, 2001, Prentice Hall, hereinafter Larman.

70. Neither Bowman, Fujikawa, nor Pyron discloses that UML class diagrams can be used to specify design information. However, Larman discloses in the "Sample UML Notation" section that UML class diagrams can be used to illustrate classes, interfaces, and their associations, i.e., that UML classes can be used to specify design information.

71. Bowman, Fujikawa, Pyron, and Larman are analogous art because they are all in the same field of endeavor of software development.

72. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman, Fujikawa, and Pyron, and the teachings of Larman before him or her to use UML class diagrams to specify the system design information. The motivation for doing so would been to use a modeling language that has become the universally accepted language for software design blueprints as specified by Larman in the Foreword of the cited reference.

73. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman in view of Fujikawa, Leonard, Pyron, as applied to claim 13 above, and further in view of Narimatsu.

74. As to claim 17, neither Bowman, Fujikawa, Leonard, nor Pyron discloses that the apparatus according to claim 13, wherein said scheduling unit describes said generated task schedule in a Gantt chart and outputs it. However, Narimatsu discloses that the apparatus according to claim 13, wherein said scheduling unit describes said generated task schedule in a Gantt chart and outputs it (Abstract, last sentence, Figures 5-13 show that the scheduling chart being output is a Gantt chart).

75. Bowman, Fujikawa, Leonard, Pyron, and Narimatsu are analogous art because they are all in the same field of endeavor. Bowman, Fujikawa, Leonard, and Pyron relate to software development, and Narimatsu relates to scheduling and project management. Scheduling and project management are pertinent to software development.

76. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bowman, Fujikawa, Leonard, Pyron, and the teachings of Narimatsu before him or her to incorporate the Gantt chart producing invention of Narimatsu in the combined system of Bowman, Fujikawa, Leonard, and Pyron. The motivation for doing so would have been the same as given above for claim 11.

77. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman in view of Fujikawa, Leonard, Pyron, as applied to claim 13 above, and further in view of Larman.

78. As to claim 18, neither Bowman, Fukijawa, Leonard, nor Pyron disclose that the

apparatus according to claim 13, wherein said system design information is a class diagram described in UML ™. However, Larman discloses that the apparatus according to claim 13, wherein said system design information is a class diagram described in UML ™ in the "Sample UML Notation" section.

79. Bowman, Fujikawa, Leonard, Pyron, and Larman are analogous art because they are all in the same field of endeavor.

80. It would have been obvious to a person having ordinary skill in the art at the time of invention having the teachings of Bomwan, Fujikawa, Leonard, Pyron, and the teachings of Larman before him or her to use UML class diagrams to specify the system design information. The motivation for doing so would been to use a modeling language that has become the universally accepted language for software design blueprints as specified by Larman in the foreword of the cited reference.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David A. Holloway whose telephone number is (571)270-1899. The examiner can normally be reached on mon-fri 8:00 am - 5:00 pm (alternate fridays off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nabil El-Hady can be reached on (571)272-3963. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DH 7/22/07

N. El-Hady

NABIL M. EL-HADY
SUPERVISORY PATENT EXAMINER